An Automatic Finite-Sample Robustness Metric: Can Dropping a Little Data Make a Big Difference?

Ryan Giordano (rgiordan@mit.edu)[1]
January 2022

[1]With coauthors Rachael Meager (LSE) and Tamara Broderick (MIT)

Dropping data: Mexico Microcredit

Example: Angelucci et al. [2015], a randomized controlled trial study of the efficacy of microcredit in Mexico based on 16,560 data points. A regression was run to estimate the average effect of microcredit.

Original result: Treatment effect statistically insignificant at 95%.

Policy implication: Disinvest in microcredit initiatives.

Data dropping: Can produce both positive and negative statististically significant results dropping no more than 15 data points ($< 0.1\%$).

Policy implication: Run a higher-powered study (not just larger N).

Cannot find influential subsets by brute force!

We provide a fast, automatic tool to approximately identify the most influential set of points.

Outline

- Why and when might you care about sensitivity to data dropping?
- How does our approximation work, and when is it accurate?

  (A formalization of the problem and the class of estimators we study.)
- Examine real-life examples of analyses: some sensitive, some not.

  (The results may defy your intuition.)
- What kinds of analyses are sensitive to data dropping?

  (Including comparison to standard errors and gross-error robustness.)

Dropping data: Motivation

Would you be concerned if you could **reverse your conclusion** by removing a **small proportion** of your data?

Not always! But sometimes, surely yes, especially when you want to **generalize to unseen, systematically different populations**.

Suppose you have a farm, and want to know whether your average yield is $>$ 170 bushels per acre. At harvest, you measure 200 bushels per acre.

- Scenario one: $>$ 170 bushels per acre means you make a profit.
  - Don't care about sensitivity to small subsets.
- Scenario two: Want to recommend methods to a distant friend.
  - Might care about sensitivity to small subsets!

Specifically, often in statistical applications:

- Policy population is different from analyzed population,
- Small fractions of data are missing not-at-random,
- We report a convenient summary (e.g. mean) of a complex effect.

Formalizing the question.

Ordinary least squares
A data point d_n has regressors x_n and response y_n: $d_n = (x_n, y_n)$.

The estimator $\hat{\theta} \in \mathbb{R}^P$ satisfies:

$$\hat{\theta} := \underset{\theta}{\arg\min} \frac{1}{2} \sum_{n=1}^{N} \left(y_n - \theta^T x_n\right)^2$$

$$\Leftrightarrow \sum_{n=1}^{N} \left(y_n - \hat{\theta}^T x_n\right) x_n = 0.$$

Make a qualitative decision using:

- A particular component: $\hat{\theta}_k$
- The end of a confidence interval: $\hat{\theta}_k + \frac{1.96}{\sqrt{N}} \hat{\sigma}(\hat{\theta})$

Z-estimators
We observe N data points $d_1, \ldots, d_N$ (in any domain).

The estimator $\hat{\theta} \in \mathbb{R}^P$ satisfies:

$$\sum_{n=1}^{N} G(\hat{\theta}, d_n) = 0_P.$$

$G(\cdot, d_n)$ is "nice," $\mathbb{R}^P$-valued.
E.g. OLS, MLE, VB, IV &c.

Make a qualitative decision using $\phi(\hat{\theta})$ for a smooth, real-valued ϕ.

(WLOG try to increase $\phi(\hat{\theta})$.)

Question: Can we make a big change in $\phi(\hat{\theta})$ by dropping $\lfloor \alpha N \rfloor$ datapoints, for some small proportion α?

Which estimators do we study?

Question: Can we make a big change in $\phi(\hat{\theta})$ by dropping $\lfloor \alpha N \rfloor$ datapoints, for some small proportion α? **Two big problems:**

- There are $\binom{N}{\lfloor \alpha N \rfloor}$ sets to check. (Huge even for $\alpha \ll 1$.)
- Evaluating $\hat{\theta}$ re-solving the estimating equation.
  - E.g., re-computing the OLS estimator.
  - Other examples are even harder (VB, machine learning)

Idea: Smoothly approximate the effect of leaving out points.

We have N data points $d_1, \ldots, d_N$, a quantity of interest $\phi(\cdot)$, and

$$\sum_{n=1}^{N} G(\hat{\theta}, d_n) = 0_P \qquad .$$

Which estimators do we study?

Question: Can we make a big change in $\phi(\hat{\theta})$ by dropping $\lfloor \alpha N \rfloor$ datapoints, for some small proportion α? **Two big problems:**

- There are $\binom{N}{\lfloor \alpha N \rfloor}$ sets to check. (Huge even for $\alpha \ll 1$.)
- Evaluating $\hat{\theta}$ re-solving the estimating equation.
  - E.g., re-computing the OLS estimator.
  - Other examples are even harder (VB, machine learning)

Idea: Smoothly approximate the effect of leaving out points.

We have N data points $d_1, \ldots, d_N$, a quantity of interest $\phi(\cdot)$, and

$$\sum_{n=1}^{N} w_n G(\hat{\theta}(w), d_n) = 0_P \text{ for a weight vector } w \in \mathbb{R}^N.$$

Which estimators do we study?

Question: Can we make a big change in $\phi(\hat{\theta})$ by dropping $\lfloor \alpha N \rfloor$ datapoints, for some small proportion α? **Two big problems:**

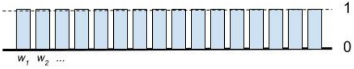
- There are $\binom{N}{\lfloor \alpha N \rfloor}$ sets to check. (Huge even for $\alpha \ll 1$.)
- Evaluating $\hat{\theta}$ re-solving the estimating equation.
    - E.g., re-computing the OLS estimator.
    - Other examples are even harder (VB, machine learning)

Idea: Smoothly approximate the effect of leaving out points.

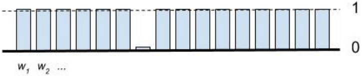
We have N data points $d_1, \ldots, d_N$, a quantity of interest $\phi(\cdot)$, and

$$\sum_{n=1}^{N} w_n G(\hat{\theta}(w), d_n) = 0_P \text{ for a weight vector } w \in \mathbb{R}^N.$$

Original weights: $\vec{1} = (1, \ldots, 1)$



Leave points out by setting their elements of w to zero.



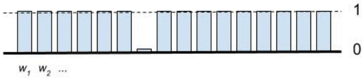
The map $w \mapsto \phi(\hat{\theta}(w))$ is well-defined even for continuous weights.

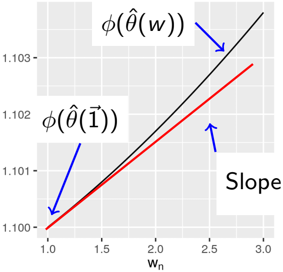
Which estimators do we study?

$$\sum_{n=1}^{N} w_n G(\hat{\theta}(w), d_n) = 0_P \text{ for a weight vector } w \in \mathbb{R}^N.$$

Original weights: $\vec{1} = (1, \ldots, 1)$

Leave points out by setting their elements of w to zero.





Slope $= \left.\frac{\partial \phi(\hat{\theta}(w))}{\partial w_n}\right|_{\vec{1}} =: \psi_n$

The values $N\psi_n$ are the **empirical influence function** [Hampel, 1986]. We call ψ_n an "influence scores."

We can use ψ_n to form a Taylor series approximation:

$$\phi(\hat{\theta}(w)) \approx \phi^{\mathrm{lin}}(w) := \phi(\hat{\theta}(\vec{1})) + \sum_{n=1}^{N} \psi_n (w_n - 1)$$

Taylor series approximation.

Problem: How much can you change $\phi(\hat{\theta}(w))$ dropping $\lfloor \alpha N \rfloor$ points? **Combinatorially hard by brute force!**

Approximate Problem: How much can you change $\phi^{\mathrm{lin}}(\hat{\theta}(w))$ dropping $\lfloor \alpha N \rfloor$ points? **Easy!**

$$\phi^{\mathrm{lin}}(w) := \phi(\hat{\theta}(\vec{1})) + \sum_{n=1}^{N} \psi_n (w_n - 1)$$

Dropped points have $w_n - 1 = -1$. Kept points have $w_n - 1 = 0$
$\Rightarrow$ The most influential points for $\phi^{\mathrm{lin}}(w)$ have the most negative ψ_n.

Procedure: (see `rgiordan/zaminfluence` on `github`)

1. Compute your original estimator $\hat{\theta}(\vec{1})$.
2. Compute and sort the influence scores $\psi_{(1)}, \ldots, \psi_{(N)}$.
3. Worry if $-\sum_{n=1}^{\lfloor \alpha N \rfloor} \psi_{(n)}$ is large enough to change your conclusions.

How to compute the ψ_n's? And how accurate is the approximation?

How to compute the influence scores?

How can we compute the influence scores $\psi_n = \left.\frac{\partial \phi(\hat{\theta}(w))}{\partial w_n}\right|_{\vec{1}}$?

By the **chain rule**, $\psi_n = \left.\frac{\partial \phi(\theta)}{\partial \theta}\right|_{\hat{\theta}(\vec{1})} \left.\frac{\partial \hat{\theta}(w)}{\partial w_n}\right|_{\vec{1}}$.

Recall that $\sum_{n=1}^{N} w_n G(\hat{\theta}(w), d_n) = 0_P$ for all w near $\vec{1}$.

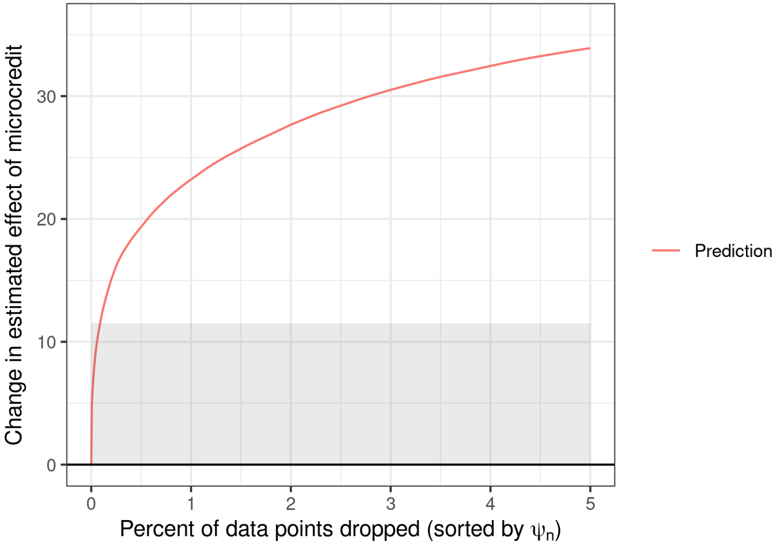
$\Rightarrow$ By the **implicit function theorem**, we can write $\left.\frac{\hat{\theta}(w)}{\partial w_n}\right|_{\vec{1}}$ as a linear system involving $G(\cdot, \cdot)$ and its derivatives.

$\Rightarrow$ The ψ_n are automatically computable from $\hat{\theta}(\vec{1})$ and software implementations of $G(\cdot, \cdot)$ and $\phi(\cdot)$ using **automatic differentiation**.

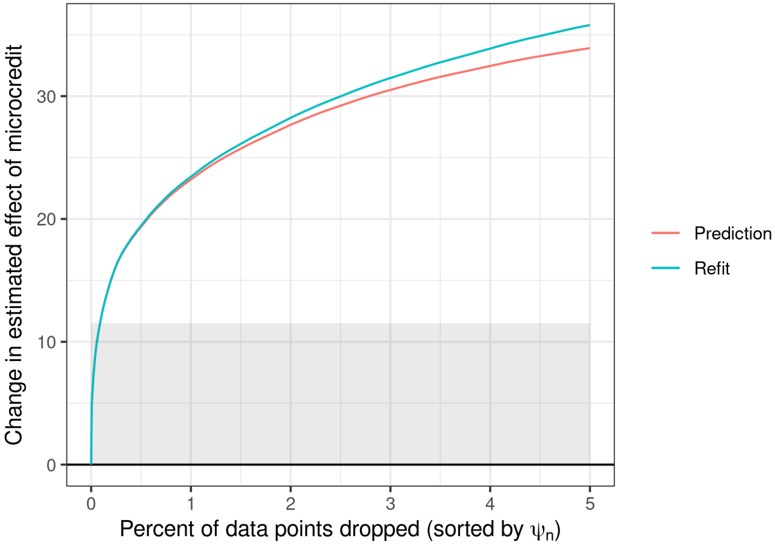
```
import jax
import jax.numpy as np
def phi(theta):
    ... computations using np and theta ...
    return value

# Exact gradient of phi (1st term in the chain rule):
jax.grad(phi)(theta_opt)
```

See `rgiordan/vittles` on `github`.

How accurate is the approximation?

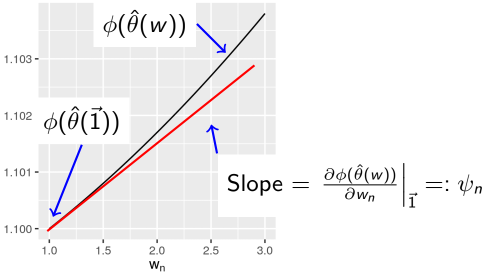


How accurate is the approximation?



How accurate is the approximation?

By conrolling the curvature, we can control the error in the linear approximation.



We provide **finite-sample theory** [Giordano et al., 2019] showing that

$$\left|\phi(\hat{\theta}(w)) - \phi^{\mathrm{lin}}(w)\right| = O\left(\left\|\frac{1}{N}(w - \vec{1})\right\|_2^2\right) = O(\alpha) \text{ as } \alpha \to 0.$$

But you don't need to rely on the theory!

Our method returns which points to drop. **Re-running once** without those points provides an **exact lower bound** on the worst-case sensitivity.

Selected experimental results.

Original estimate (SE)	Refit estimate (SE)	Observations dropped
-4.549 (5.879)	7.030 (2.549)*	$15 = 0.09\%$

Table: Microcredit Mexico results [Angelucci et al., 2015].

A $*$ indicates statistical significance at the 95% level.

Selected experimental results.

Original estimate (SE)	Refit estimate (SE)	Observations dropped
-4.549 (5.879)	7.030 (2.549)*	$15 = 0.09\%$

Table: Microcredit Mexico results [Angelucci et al., 2015].

Original estimate (SE)	Refit estimate (SE)	Observations dropped
33.861 (4.468)*	-9.416 (3.296)*	$986 = 9.37\%$

Table: Cash transfers results. [Angelucci and De Giorgi, 2009]

A $*$ indicates statistical significance at the 95% level.

Selected experimental results.

Original estimate (SE)	Refit estimate (SE)	Observations dropped
-4.549 (5.879)	7.030 (2.549)*	15 = 0.09%

Table: Microcredit Mexico results [Angelucci et al., 2015].

Original estimate (SE)	Refit estimate (SE)	Observations dropped
33.861 (4.468)*	-9.416 (3.296)*	986 = 9.37%

Table: Cash transfers results. [Angelucci and De Giorgi, 2009]

Original estimate (SE)	Refit estimate (SE)	Observations dropped
0.029 (0.005)*	-0.009 (0.004)*	224 = 0.96%

Table: Medicaid profit results [Finkelstein et al., 2012]

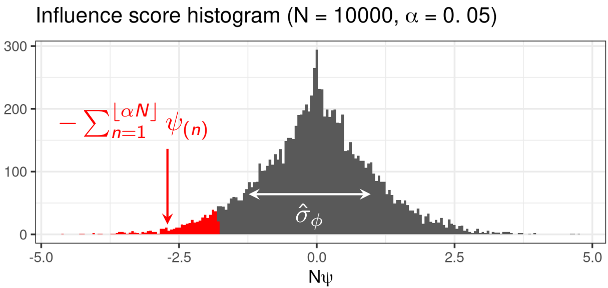
A $*$ indicates statistical significance at the 95% level.

What makes an analysis sensitive?

We are "sensitive to data dropping" if, for some Δ large enough to change conclusions, $\exists w^*$ dropping $\lfloor \alpha N \rfloor$ points such that

$$\text{“Signal”} := \Delta < \phi^{\mathrm{lin}}(w^*) - \phi(\hat{\theta}(\vec{1})) = -\sum_{n=1}^{\lfloor \alpha N \rfloor} \psi_{(n)} =: \hat{\sigma}_\phi \hat{\mathscr{T}}_\alpha$$

- The "noise" $\hat{\sigma}_\phi^2 \to \mathrm{Var}(\sqrt{N}\phi)$ ("sandwich" variance estimator)
- The "shape" $\hat{\mathscr{T}}_\alpha := \frac{-\sum_{n=1}^{\lfloor \alpha N \rfloor} \psi_{(n)}}{\hat{\sigma}_\phi} \to$ nonzero constant $\leq \sqrt{\alpha(1-\alpha)}$



Example.

$$\alpha := \text{Proportion of points to drop}$$
$$\Delta := \text{Signal (difference large enough to change conclusions)}$$
$$\hat{\sigma}_\phi := \text{Noise (consistent estimator of Var}\left(\sqrt{N}\phi\right)\text{)}$$
$$\hat{\mathscr{T}}_\alpha := \text{Shape (bounded by } \sqrt{\alpha(1-\alpha)} \text{ and given by } N\psi_n \text{ tail shape)}$$

Sensitive to data dropping if:

$$\phi^{\mathrm{lin}}(w^*) - \phi(\hat{\theta}(\vec{1})) = \hat{\sigma}_\phi \hat{\mathscr{T}}_\alpha \geq \Delta \qquad \Leftrightarrow \qquad \frac{\Delta}{\hat{\sigma}_\phi} \leq \hat{\mathscr{T}}_\alpha.$$

The **signal to noise ratio** $\frac{\Delta}{\hat{\sigma}_\phi}$ determines sensitivity to data dropping.

Contrast with standard errors. A 95% CI is given by $\phi(\hat{\theta}(\vec{1})) \pm \frac{1.96}{\sqrt{N}}\hat{\sigma}_\phi$. We fail to reject the value $\phi(\hat{\theta}(\vec{1})) + \Delta$ when

$$\phi(\hat{\theta}(\vec{1})) + \Delta \leq \phi(\hat{\theta}(\vec{1})) + \frac{1.96}{\sqrt{N}}\hat{\sigma}_\phi \qquad \Leftrightarrow \qquad \frac{\Delta}{\hat{\sigma}_\phi} \leq \frac{1.96}{\sqrt{N}}.$$

Corollaries.

Robust to data dropping:
("dropping robustness")

$$\mathrm{SNR} = \frac{\Delta}{\hat{\sigma}_\phi} > \hat{\mathscr{T}}_\alpha$$

Robust to sampling variation:
("sampling robustness")

$$\mathrm{SNR} = \frac{\Delta}{\hat{\sigma}_\phi} > \frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$$

- **Dropping robustness $\neq$ sampling robustness in general.**

Proof: $\hat{\mathscr{T}}_\alpha \neq \frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$.

- **When the SNR is small, sufficiently large N produces sampling robustness, but not necessarily dropping robustness.**

Proof: $\frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi \rightarrow 0$, but $\hat{\mathscr{T}}_\alpha \rightarrow$ a nonzero constant.

- **Statistical insignificance is dropping non-robust for large N.**

Proof: Insignificance means $|\phi(\hat{\theta}(\vec{1}))| \leq \frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$.

$\Rightarrow$ A result can be made significant by a change of no more than $\frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$.

$\Rightarrow$ The SNR for a conclusion of "insignificance" is $\frac{\Delta}{\hat{\sigma}_\phi} \leq \frac{1.96}{\sqrt{N}} \rightarrow 0 \leq \hat{\mathscr{T}}_\alpha$.

Corollaries.

Robust to data dropping:
("dropping robustness")

$$\mathrm{SNR} = \frac{\Delta}{\hat{\sigma}_\phi} > \hat{\mathscr{T}}_\alpha$$

Robust to gross errors:
("gross error robustness")

Gross outliers cannot produce arbitrarily large changes to ϕ.

- **Dropping non-robustness is not driven by misspecification.**

Proof: Small Δ are dropping non-robust irrespective of specification.

- **Gross outliers primarily affect dropping robustness through $\hat{\sigma}_\phi$.**

Proof: For a fixed $\hat{\sigma}_\phi$, outliers decrease $\hat{\mathscr{T}}_\alpha$. (Details in paper.)

- **To achieve dropping robustness, reduce $\hat{\sigma}_\phi$ and / or increase Δ.**

Proof: Across typical distributions, $\hat{\mathscr{T}}_\alpha$ varies litte. (Details in paper.)

Conclusion

- You may be concerned if you could reverse your conclusion by removing a small proportion of your data.

Conclusion

- You may be concerned if you could reverse your conclusion by removing a small proportion of your data.
- We can quickly and automatically find an approximate influential set which is accurate for small sets.

Conclusion

- You may be concerned if you could reverse your conclusion by removing a small proportion of your data.
- We can quickly and automatically find an approximate influential set which is accurate for small sets.
- Data dropping robustness is principally determined by the signal to noise ratio, and captures sensitivity distinct from sampling and gross error sensitivity.

Links and references

Tamara Broderick, Ryan Giordano, Rachael Meager (alphabetical authors)
"An Automatic Finite-Sample Robustness Metric: Can Dropping a Little Data Change Conclusions?"
https://arxiv.org/abs/2011.14999

Blog posts with more details:

- Colinearity in OLS after dropping
- Connections to the bootstrap
- Data dropping sensitivity overcomes p-hacking
- When a norm is the quantity of interest

Related software on `github`:

- rgiordan/zaminfluence (for `R`)
- rgiordan/vittles (for `Python`)

Some of my work on other forms of robustness:

- Prior sensitivity in Bayesian nonparametrics [Giordano et al., 2021]
- Model sensitivity of MCMC output [Giordano et al., 2018]
- Cross-validation [Giordano et al., 2019]
- Frequentist variances of MCMC posteriors (in progress)

References

M. Angelucci and G. De Giorgi. Indirect effects of an aid program: How do cash transfers affect ineligibles' consumption? *American Economic Review*, 99(1):486–508, 2009.

M. Angelucci, D. Karlan, and J. Zinman. Microcredit impacts: Evidence from a randomized microcredit program placement experiment by Compartamos Banco. *American Economic Journal: Applied Economics*, 7(1):151–82, 2015.

A. Baydin, B. Pearlmutter, A. Radul, and J. Siskind. Automatic differentiation in machine learning: A survey. *The Journal of Machine Learning Research*, 18(1):5595–5637, 2017.

A. Finkelstein, S. Taubman, B. Wright, M. Bernstein, J. Gruber, J. Newhouse, H. Allen, K. Baicker, and Oregon Health Study Group. The Oregon health insurance experiment: Evidence from the first year. *The Quarterly Journal of Economics*, 127(3):1057–1106, 2012.

R. Giordano, T. Broderick, and M. I. Jordan. Covariances, robustness and variational Bayes. *The Journal of Machine Learning Research*, 19(1):1981–2029, 2018.

R. Giordano, W. Stephenson, R. Liu, M. I. Jordan, and T. Broderick. A swiss army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147. PMLR, 2019.

R. Giordano, R. Liu, M. I. Jordan, and T. Broderick. Evaluating sensitivity to the stick-breaking prior in Bayesian nonparametrics. 2021.

P. Gustafson. Local robustness in Bayesian analysis. In *Robust Bayesian Analysis*, pages 71–88. Springer, 2000.

F. Hampel. *Robust statistics: The approach based on influence functions*, volume 196. Wiley-Interscience, 1986.

A. Wilson, M. Kasy, and L. Mackey. Approximate cross-validation: Guarantees for model assessment and selection. In *International Conference on Artificial Intelligence and Statistics*, pages 4530–4540. PMLR, 2020.